REMARKS

Claims 1-36 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Provisional Double Patenting Rejection:

The Examiner provisionally rejected claims 1-36 under the judiciary created doctrine of double patenting over claims 1-56 of copending Application No. 10/618,810. If this rejection becomes non-provisional, Applicants will either submit a terminal disclaimer or present arguments traversing the rejection.

Section 103(a) Rejections:

The Examiner rejected claims 1-8 and 10-14 under 35 USC § 103(a) as being unpatentable over Hagersten et al. (U.S. Patent 5,983,326) (hereinafter "Hagersten") in view of Klein et al. (U.S. Patent 6,728,958) (hereinafter "Klein"), claim 9 as being unpatentable over Hagersten in view of Klein and further in view of Fowler (U.S. Patent 4,502,116), claims 15-19, 25-30 and 36 as being unpatentable over Hagersten in view of Fowler, and claims 20-24 and 31-35 as being unpatentable over Hagersten in view of Fowler and further in view of Klein. Applicants respectfully traverse these rejections for at least the following reasons.

Claim 1

Regarding claim 1, contrary to the Examiner's assertion, Hagersten in view of Klein fails to disclose, *memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement a transaction manager*. The Examiner refers to the fact that Hagersten describes memory that includes code for use by its processors. However, the code in Hagersten's memory is not described as executable by the one or more processors to implement a transaction

manager. In fact, the functionality in Hagersten relied upon by the Examiner to teach the transaction manager of claim 1 is part of system interface 24 which is separate from memory 22 and processors 16. *See* Hagersten, col. 7, lines 44-53 cited by the Examiner. Hagersten's home agent and system interface is not described as being implemented by program instructions stored in memory 22.

Further regarding claim 1, contrary to the Examiner's assertion, Hagersten in view of Klein fails to disclose, *a transaction manager configured to manage a plurality of transactions initiated by one or more applications*. The Examiner relies on Klein to teach the transaction being initiated by one or more applications. However, the cited portion of Klein only describes an application initiating a single transaction. Therefore, the Examiner's assertion is not supported by the cited art.

Further regarding claim 1, contrary to the Examiner's assertion, Hagersten in view of Klein fails to disclose, *wherein each transaction comprises a plurality of operations to one or more data sources that are required to be committed to the one or more data sources atomically for each respective transaction*. The Examiner refers to col. 8, lines 20-34, of Hagersten. However, Hagersten does not teach that each transaction comprises a plurality of operations. To the contrary, the portion of Hagersten cited by the Examiner states that a transaction includes at most a single memory operation. Also, the portion of Hagersten cited by the Examiner says nothing of each transaction comprising a plurality of operations that are required to be committed to the one or more data sources atomically for each respective transaction.

Further regarding claim 1, contrary to the Examiner's assertion, Hagersten in view of Klein fails to disclose, *pause the plurality of transactions managed by the transaction manager **in response to a pause request** to pause the transaction manager, wherein while paused, the transaction manager does not allow any of the plurality of transactions managed by the transaction manager to complete*. The Examiner cites Hagersten, column 5, lines 55-64; column 7, lines 44-53, and column 17, lines 7-9. These passages teach one or more queues configured to receive transaction requests from

processing nodes. The passages also teach a home agent control unit configured to receive and service transaction requests and a transaction blocking unit coupled to the queues and the home agent. For example, column 5, lines 60 through column 6, line 4 states:

> The transaction blocking unit is configured to block selected transactions if another transaction request to a common coherency unit is currently being serviced by the home agent control unit. The transaction blocking unit is further configured to allow servicing of a given transaction request to a particular coherency unit if a second transaction request to the particular coherency unit is currently being serviced by the home agent control unit and if the second transaction request does not cause ownership of the particular coherency unit and if the second transaction request and the given transaction request are the same transaction type.

Applicants assert that the transaction blocking unit of Hagersten clearly does not block transactions **in response to a pause request** as required by Applicants' claim 1. Instead, the transaction blocking unit, blocks a second transaction if a first transaction request (to a common coherency unit) is currently being serviced by the home agent. The transaction blocking unit is not **responding to a request** to pause, but rather is responding to the detection of a second transaction request arriving when the home agent is servicing a first transaction request. Nowhere does Hagersten or Klein, taken in combination or singularly, teach *a transaction manager configured to pause the plurality of transactions managed by the transaction manager* **in response to a pause request.**

Further in regard to claim 1, Hagersten in view of Klein fails to disclose *wherein while paused, the transaction manager* **does not allow any of the plurality of transactions** *managed by the transaction manager to complete.* As described in the quotation above, Hagersten teaches the blocking of one transaction request while another transaction request is being serviced. Hagersten states in column 7, lines 46-49, "However, a blocking mechanism is employed to prevent the servicing of a particular coherent transaction request if another transaction request corresponding to the same coherency unit is currently being service by the system interface." Hagersten, whether or not combined with Klein, does not teach pausing **all** transactions managed by the transaction manager in response to a pause request as required by Applicants' claim 1.

To the contrary, Hagersten <u>explicitly</u> allows the current transaction to complete while blocking one or more other transactions.

Further in regard to claim 1, contrary to the Examiner's assertion, Hagersten in view of Klein fails to disclose that *the transaction manager is configured to resume the plurality of transactions managed by the transaction manager in response to a resume request.* The Examiner cites column 17, lines 9-12, which states, "...subsequent requests involving the coherency unit are not performed until the coherency activity corresponding to the coherency request is completed." Applicants assert that Hagersten does not teach resuming the transaction manager <u>in response to a resume request</u>. Instead, Hagersten teaches activity is resumed when the coherency unit completes the current task. Clearly there is no resume <u>request</u> taught by Hagersten. Further, the Examiner admits Hagersten fails to teach generating a pause request and a resume request in paragraph 24 of the Instant Office Action. The Examiner states, "Hagersten does not explicitly teach generating a pausing request and a resuming request." In addition, Klein fails to teach a pause request or a resume request.

Claim 14

Regarding claim 14, contrary to the Examiner's assertion, Hagersten in view of Klein fails to disclose, *memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement one or more application servers, wherein each one or more <u>application servers</u> is configured to <u>run one or more applications</u> ...and <u>provide one or more transaction managers</u>.* The Examiner refers to the fact that Hagersten describes memory that includes code for use by its processors. However, the code in Hagersten's memory is not described as executable by the one or more processors to implement an application server(s) that runs an application(s) and provides a transaction manager(s). In fact, Hagersten has absolutely nothing to do with application servers. Furthermore, the functionality in Hagersten relied upon by the Examiner to teach the one or more transaction managers of claim 14 is part of system interface 24 which is separate from memory 22 and processors 16. *See*

Hagersten, col. 7, lines 44-53 cited by the Examiner. Hagersten's home agent and system interface is not described as being implemented by program instructions stored in memory 22.

Further regarding claim 14, contrary to the Examiner's assertion, Hagersten in view of Klein fails to disclose, *wherein each of the one or more transactions comprises a plurality of operations to one or more data sources that are required to be committed to the one or more data sources atomically for each respective transaction*. The Examiner refers to col. 8, lines 20-34, of Hagersten. However, Hagersten does not teach that each transaction comprises a plurality of operations. To the contrary, the portion of Hagersten cited by the Examiner states that a transaction includes at most a single memory operation. Also, the portion of Hagersten cited by the Examiner says nothing of each transaction comprising a plurality of operations that are required to be committed to the one or more data sources atomically for each respective transaction.

Further regarding claim 14, contrary to the Examiner's assertion, Hagersten in view of Klein fails to disclose that *one of the transaction managers is configured to pause a corresponding one or more transactions in response to a pause request*. The Examiner cites Hagersten, column 5, lines 55-64; column 7, lines 44-53, and column 17, lines 7-9. These passages teach one or more queues configured to receive transaction requests from processing nodes. The passages also teach a home agent control unit configured to receive and service transaction requests and a transaction blocking unit coupled to the queues and the home agent. For example, column 5, lines 60 through column 6, line 4 states:

> The transaction blocking unit is configured to block selected transactions if another transaction request to a common coherency unit is currently being serviced by the home agent control unit. The transaction blocking unit is further configured to allow servicing of a given transaction request to a particular coherency unit if a second transaction request to the particular coherency unit is currently being serviced by the home agent control unit and if the second transaction request does not cause ownership of the particular coherency unit and if the second transaction request and the given transaction request are the same transaction type.

Applicants assert that the transaction blocking unit of Hagersten clearly does not block transactions **in response to a pause request** as required by Applicants' claim 14. Instead, the transaction blocking unit, blocks a second transaction if a first transaction request (to a common coherency unit) is currently being serviced by the home agent. The transaction blocking unit is not **responding to a request** to pause, but rather is responding to the detection of a second transaction request arriving when the home agent is servicing a first transaction request. Nowhere does Hagersten or Klein, taken in combination or singularly, teach *a transaction manager configured to pause a corresponding one or more transactions* **in response to a pause request.**

Further in regard to claim 14, contrary to the Examiner's assertion, Hagersten in view of Klein fails to disclose that *the transaction manager is configured to resume the corresponding one or more transactions in response to a resume request*. The Examiner cites column 17, lines 9-12, which states, "…subsequent requests involving the coherency unit are not performed until the coherency activity corresponding to the coherency request is completed." Applicants assert that Hagersten does not teach resuming a transaction in response to a resume request. Instead, Hagersten teaches activity is resumed when the coherency unit completes the current task. Clearly there is no resume request taught by Hagersten. Further, the Examiner admits Hagersten fails to teach generating a pause request and a resume request in paragraph 24 of the Instant Office Action. The Examiner states, "Hagersten does not explicitly teach generating a pausing request and a resuming request." In addition, Klein fails to teach a pause request or a resume request.

Claim 15

Regarding claim 15, contrary to the Examiner's assertion, Hagersten in view of Fowler fails to disclose, *wherein each transaction managed by the transaction manager comprises a plurality of operations to one or more data sources that are required to be committed to the one or more data sources atomically for each respective transaction*. The Examiner refers to col. 8, lines 20-34, of Hagersten. However, Hagersten does not

teach that each transaction comprises a <u>plurality</u> of operations. To the contrary, the portion of Hagersten cited by the Examiner states that a transaction includes at most a <u>single</u> memory operation. Also, the portion of Hagersten cited by the Examiner says nothing of each transaction comprising a <u>plurality</u> of operations that are required to be committed to the one or more data sources <u>atomically</u> for each respective transaction.

Further regarding claim 15, the Examiner admits Hagersten fails to teach, *generating a request to pause a transaction manager*, and *generating a request to resume the transaction manager*, and the Examiner relies on Fowler, column 5, lines 40-53 and column 8, lines 52-56. These passages teach a synchronization interface circuit that provides users with toggle switches to pause and resume the <u>synchronization of a multiprocessor system</u>. Column 5, lines 43-47 state, "The control circuitry provides a manual pause switch to allow a user to manually generate a pause signal. This allows the user to stop or pause the subsystems by a means other than the test utilities." A toggle switch used to pause and resume one or more processors <u>has absolutely nothing to do with</u> generating requests to pause and resume a transaction manager managing multiple transactions. In fact, in Fowler, the human user generates the request to pause and resume the processors by manually manipulating the switch. The system as taught by Fowler merely responds to the request. Further, a <u>manual toggle switch</u> used to manually pause and resume one or more processors has nothing to do with a transaction manager.

Combining Hagersten and Fowler would merely result in the system that would block pending coherent transaction requests if another transaction request corresponding to the same coherency unit is already being serviced, and include physical toggle switches (manipulated by a human user) for pausing and resuming the processors for test purposes. Such a combination would clearly not result in Applicants' invention as recited in claim 15.

Further in regard to claim 15, Hagersten and Fowler do not teach pausing and resuming the transaction manager *in response to a request*. As discussed above in regard

to claims 1 and 14, the home agent in Hagersten blocks transaction based on detecting the state of an existing transaction, **not in response to any <u>request</u>**.

<u>Claim 26</u>:

Similar arguments as presented above for claim 15 apply to claim 26 as well.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejections have been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-15100/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: ___January 29, 2008___